

# Documentation for Q1 part A and B

## Q1 part A)

Have created 3 functions `countA()` , `countB()` , `countC()` that count from 0 to  $(1-2^{32})$  and they are being called by 3 new threads that being created in the main function.

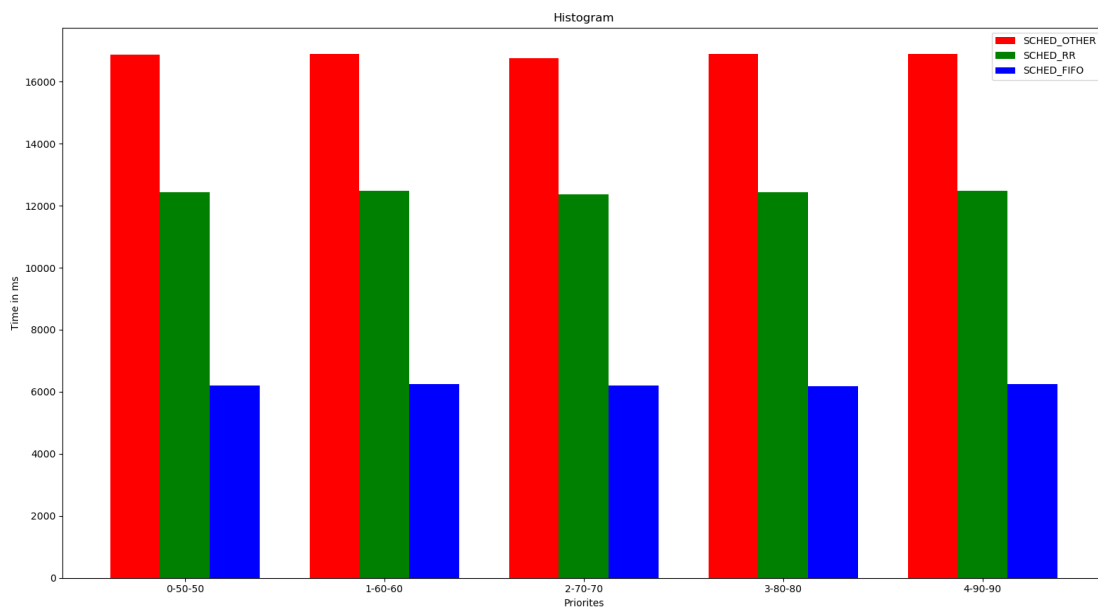
Each thread has its own scheduling priority, `SCHED_OTHER`, `SCHED_RR`, `SCHED_FIFO`.

Have used `clock_gettime()` function to return the running times of each of the threads.

All threads run parallelly and have the core affinity set to 0.

The findings all have the similar result of `SCHED_OTHER` being the slowest while `SCHED_FIFO`.

I have plotted the graph for the same using the `histogram.py` that plots the values that are received from the function



## Q1 part B)

Created 3 processes with the help of `pid_t` and calculated its time.

Applied the idea of forking using `fork()` and executed the processes using `execvp()`

```
BUILD arch/x86/boot/bzImage
Kernel: arch/x86/boot/bzImage is ready (#1)

real    13m37.340s
user    22m30.533s
sys      7m16.551s

BUILD arch/x86/boot/bzImage
Kernel: arch/x86/boot/bzImage is ready (#1)

real    9m44.129s
user    21m37.940s
sys      8m53.973s
CLEAN arch/x86/boot/compressed

BUILD arch/x86/boot/bzImage
Kernel: arch/x86/boot/bzImage is ready (#1)

real    9m34.070s
user    22m9.555s
sys     10m46.804s
[Dhvanil submission3]#
```